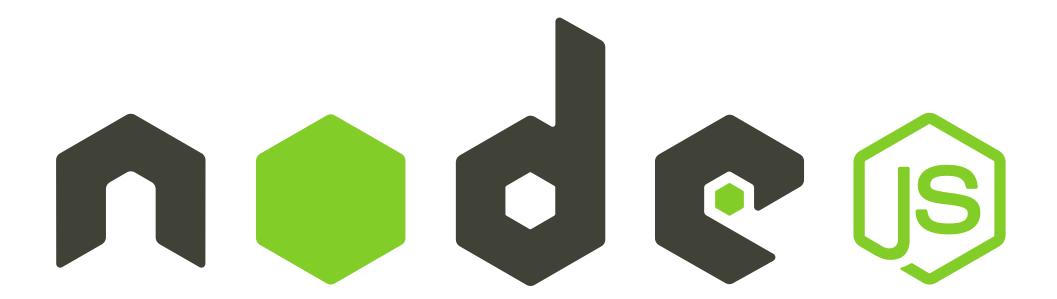
Muhammad Rafaqat

@codewithrafaqat



Complete Models

FUNDAMentals

- Event-Driven Architecture: Uses events and callbacks for asynchronous operations.
- Non-Blocking I/O: Doesn't wait for operations to complete before moving to the next task.
- Global Objects: process, console, Buffer,
 __dirname.
- Modules: require (CommonJS) vs import (ES Modules).

ASYNCHRONOUS PROGRAMMING

- Callbacks: Functions passed as arguments to handle async results.
- Promises: Handle async operations with .then() and .catch().
- Async/Await: Syntactic sugar over Promises for cleaner code.
- Event Loop: Mechanism that handles async callbacks in phases (timers, I/O, etc.).

FILE SYSTEM & STREAMS

- fs Module: Read/write files synchronously or asynchronously.
- Streams: Handle large data chunks efficiently (Readable, Writable, Pipe).

Networking

- HTTP/HTTPS Modules: Create servers and make requests.
- RESTful APIs: Design endpoints for CRUD operations.
- WebSockets: Real-time communication (socket.io or ws).

Debugging & Performance

- --inspect Flag: Debug with
 Chrome DevTools.
- Clustering: Utilize multi-core
 CPUs with the cluster module.
- Memory Leaks: Identify using heap snapshots.



SECURITY

- Helmet.js: Secure HTTP headers.
- Input Validation: Sanitize user inputs to prevent attacks.
- Rate Limiting: Prevent bruteforce attacks.

DATABASES

- MongoDB: NoSQL database with Mongoose ODM.
- SQL: PostgreSQL/MySQL with Sequelize or TypeORM.
- Redis: Caching and real-time features.

Testing

- Unit Tests: Jest/Mocha for isolated function testing.
- Integration Tests: Supertest
 for API endpoints.



Interview Questions

Beginner-Level Questions

- 1. What is Node.js and how does it work?
- 2. What are the key features of Node.js?
- 3. What is the difference between synchronous and asynchronous functions in Node.js?
- 4. What is npm and how is it used?
- 5. How do you create a simple server in Node.js using the HTTP module?

Intermediate Level Questions

- 1. What are streams in Node.js and how are they used?
- 2. What is the Event Loop in Node.js and how does it work?
- 3. Explain the concept of middleware in Node.js (especially with Express.js).
- 4. How does Node.js handle errors? What are the best practices for error handling?
- 5. What is the difference between process.nextTick(), setImmediate(), and setTimeout() in Node.js?

ADVANCE LEVEL QUESTIONS

- 1. What are worker threads in Node.js and when should you use them?
- 2. How does clustering work in Node.js and why is it used?
- 3. What is memory leak in Node.js and how can you detect and prevent it?
- 4. How would you secure a Node.js application?

 (Mention techniques like Helmet, Rate Limiting, etc.)
- 5. Explain how to handle high-performance, real-time data in Node.js (e.g., WebSockets, Redis, etc.).

If you found this content useful, don't forget to like, comment, and share it with your network.



